BASTIAN: incorporating the Rough Sets theory into a Case-Based Classifier System

Maria Salamó i Llorente mariasal@salleURL.edu
Elisabet Golobardes i Ribé elisabet@salleURL.edu

Grup de Recerca en Sistemes Intel·ligents
Departament d'Informàtica
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull (URL)
Passeig Bonanova 8 08022-Barcelona

Abstract

This paper proposes how to incorporate the Rough Sets theory as a weighting method into a Case-Based Classifier System. This hybrid system has been implemented into the platform called BASTIAN (case-**BA**sed **S**ys**T**em **I**n cl**A**ssificatio**N**), which incorporate both techniques.

Thus, the main goals of the paper are: presenting the BASTIAN system, describing the hybrid method; and analysing this proposal for different domains, extracted from the UCI repository.

Keywords: Case-Based Reasoning, Machine Learning, Diagnose, Knowledge Discovery

1 Introduction

Our main goal is to develop, evaluate and improve the classifier systems. In this paper we present a hybrid classifier system based on *Case-Based Reasoning* and *Rough Sets*. The BASTIAN platform is a Case-Based Reasoning system that incorporates Rough Sets capabilities in order to improve the prediction accuracy rate. Rough Sets theory is used in our system as a weighting method to select the best feature relevance of the domain.

Case-Based Reasoning (CBR)[1] have been used in a wide variety of fields and applications. We use CBR as an automatic classification system [4, 21].

Rough Sets theory is a Data Mining technique. The nature of Rough Sets theory has made them useful for reducing the knowledge, extracting dependencies in knowledge, reasoning about knowledge, pattern recognition, etc.

The main research trends in Rough Sets theory -which tries to extend the capabilities of reasoning systems- are:

1. The treatment of incomplete knowledge.
2. The management of inconsistent pieces of information.
3. The manipulation of various levels of representation, moving from refined universes of discourse to coarser ones and conversely.

The paper is structured as described. First, an overview about the BASTIAN platform in section 2. Next section proposes the Rough Sets theory as a weighting method for a Case-Based classifier system. Sections 4 and 5 expose the testbed used and the results obtained respectively. Finally, the last section presents the conclusions and further work.

2 BASTIAN System description

BASTIAN platform is a Case-Based Reasoning system used in classification. Case-Based Reasoning integrates in one system two different characteristics: machine learning capabilities and problem solving capabilities. CBR uses a similar philosophy to that which humans sometimes use: it tries to solve new cases (examples) of a problem by using old previously solved cases [16]. The process of solving new cases contributes with new information and new knowledge to the system. This new information can be used for solving other future cases. The basic method, see Figure 1, can be easily described in terms of its four phases [1, 11]:

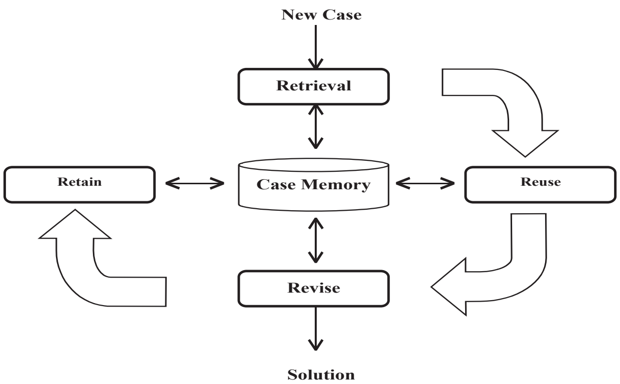

Figure 1: CBR Cycle.

The first phase retrieves old solved cases similar to the new one. In the second phase, the system tries to reuse the solutions of the previously retrieved cases for solving the new case. The third phase revises the proposed solution. Finally, the fourth phase retains the useful information obtained when solving the new case. In a Case-Based Classifier System, it is possible to simplify the reuse phase classifying the new case with the same class as the most similar retrieved case.

BASTIAN system is an extension of CaB-CS (**Ca**se-**B**ased **C**lassifier **S**ystem) system [9, 7, 6]. It allows the user to test several variants of CBR. To be exact, the variant presented in this paper is focused on two different phases: the retrieval and the retain phase, and also on the case memory organisation. BASTIAN has been developed in JAVA language and the system is being improved with new capabilities.

2.1 General Structure

The BASTIAN general structure, see figure 2, maintains the four phases described in [1]. The system adds a previous phase *StartupInterface*, not incorporate on the Case-Based Reasoning cycle, that prepares the initial start-up of the system.

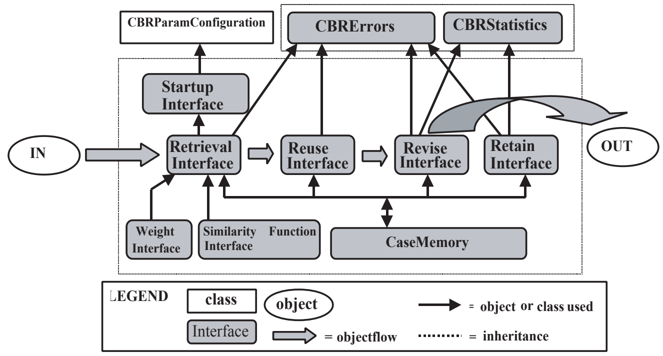

Figure 2: General Structure in BASTIAN.

The system functionalities are developed to work separately and independent in co-operation among the rest. Each functionality described in the general structure has a description of the general behaviour that has to achieve. The main goal is to obtain a general structure that could change dynamically depending on the type of Case-Based Reasoner we want to develop. The main functionalities are:

- The *CBRParamConfiguration* allows us to change and get the configuration. The configuration could be changed independent of the system, this means that is not necessary to execute the system in order to change the configuration and it can be changed during the CBR cycle too.
- The *CBRErrors* is the error control functionality which detects all the possible problems during one execution.
- The *CBRStatistics* aims to develop all the possible statistics during execution of the system. It computes the statistics in EXCEL, LATEX and EPS format.
- The *CaseMemory* goal is to develop different case memory organisations.
- The *SimilarityFunctionInterface* concentrates all the characteristics related to similarity functions. It let us change the similarity function dynamically into the system during one execution.
- The *WeightingInterface*, contains the main abilities to compute the feature relevance in a Case-Based Classifier System. It is related to

the *RetrievalInterface* and the *SimilarityFunctionInterface*.

- The {*Retrieval, Reuse, Revise, Retain*}*Interface* are the four phases of the CBR cycle. These interfaces describe the behaviour of each phase.

The kernel in a Case-Based Reasoning system is the retrieval phase (phase 1). Phase 1 retrieves the most similar case or cases to the new case. Obviously, the meaning of most similar will be a key concept in the whole system. Similarity between two cases is computed using different similarity functions. Our aim is to improve this similarity functions accuracy using a weighting method that computes automatically the feature relevance [2, 5, 10].

2.2 Similarity Functions

For our purpose in this paper, we use the similarity functions based on the distance concept introduced in BASTIAN. The most used similarity function is the Nearest Neighbour algorithm, which computes the similarity between two cases using a global similarity measure [2, 3]. The practical implementation (used in our system) of this function is based on the*Minkowsky's metric* [6, 12] and we also use the *Clark's distance* and the *Cosine distance* [17].

2.2.1 Minkowsky's metric

The Minkowsky's metric is defined as:

$$Sim(Case\_x, Case\_y) = \sqrt[r]{\sum_{i=1}^{F} w_i \times |x_i - y_i|^r} \quad (1)$$

Where *Case_x* and *Case_y* are two cases, whose similarity is computed; F is the number of features that describes the case; x_i, y_i represent the value of the *ith* feature of cases *Case_x* and *Case_y* respectively; and w_i is the weight of the *ith* feature.

In this study we test the Minkowsky's metric for three different values of r: *Hamming distance* for $r = 1$, *Euclidean distance* for $r = 2$, and *Cubic distance* for $r = 3$.

2.2.2 Clark's distance

The Clark's distance is defined as:

$$Sim(Case\_x, Case\_y) = \sqrt[2]{\sum_{i=1}^{F} w_i \cdot \frac{|(x_i - y_i)|^2}{|(x_i + y_i)|^2}} \quad (2)$$

Where $Case\_x$ and $Case\_y$ are two cases, whose similarity is computed; F is the number of features that describes the case; and x_i, y_i represent the value of the ith feature of cases $Case\_x$ and $Case\_y$ respectively; and w_i is the weight of the *ith* feature.

2.2.3 Cosine distance

The Cosine distance is based on vector properties in an Euclidean space. It measures the Cosine angle in a n-dimensional vector space. This metric is defined as:

$$Sim(Case\_x, Case\_y) = w_i \cdot \frac{\sum_{i=1}^{F} (x_i \cdot y_i)}{\sqrt[2]{(\sum_{i=1}^{F} x_i^2) \cdot (\sum_{i=1}^{F} y_i^2)}} \quad (3)$$

Where F represents the number of features that describe the cases; and x_i, y_i represent the value of the *ith* feature of cases $Case\_x$ and $Case\_y$ respectively; and w_i is the weight of the *ith* feature.

2.3 Memory Representation

The case memory structure is specified in figure 3. As it can be seen, there are three structures that can be used in BASTIAN: the first one is a *list*, the second one is a *SingleList* (a vector) and the last one is a *tree*. The memory representation used in the experiments has been the first one, a list of cases. The second part of the figure 3 shows the problems that we have used in this work.

The representation used in each sample is based on an attribute-value representation, see equation 4.

$$Case = \{a_0, a_1, a_2, \cdots, a_n, CLASS\} \quad (4)$$

Where a_i are the value for the attribute i; and $CLASS$ is the class that the case belongs to.

2.4 Retain Policies

In order to decide whether a case is representative enough to be stored in the case memory, we use three different policies, see figure 4:

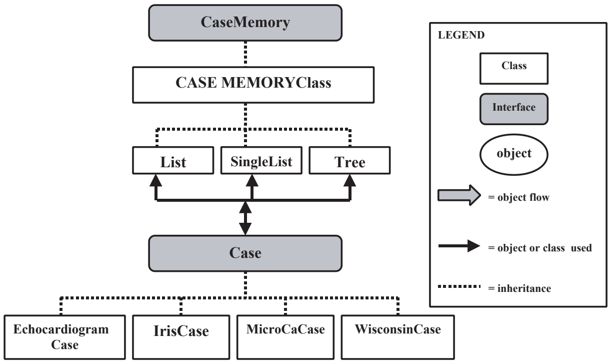

Figure 3: Case Memory Structure in BASTIAN.

- **Test mode**, in this mode system does not store any new case in the case memory. This criterion has been used for two reasons. On one hand, the results obtained using this mode can be compared, in equal conditions, to those obtained using other machine learning methods that do not include learning while solving new problems. On the other hand, it allows us to evaluate the initial *corpus* of the case memory.

- **DifSim mode**, under this policy the system stores the new case if its similarity with the retrieved case is not zero. In other words, the new case will be stored if there is not any identical case in the case memory.

- **DifClass mode**, this is an intermediate solution between the previous ones. The system will store the new case if it has been impossible to classify it correctly. Otherwise, it will not be stored.

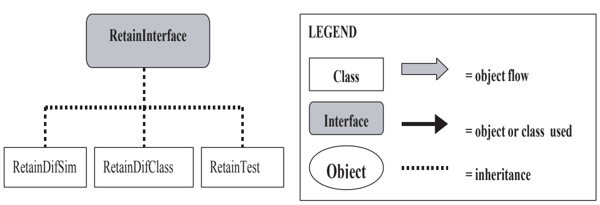

Figure 4: Retain Structure in BASTIAN.

The system let us also to train the initial case memory to store only the most representative cases.

3 Feature Relevance

BASTIAN includes 3 variants to weight the feature relevance. The first one is the Sample Correlation [9]; the second one is the Shannon Entropy [13]; and the third is the Rough Sets theory [18]. The aim of this paper is to explain the integration of the third one into the BASTIAN system. The Rough Sets into the BASTIAN system can be applied using two policies:

- **Static:** we compute the weight of the features only using the initial case memory. Our paper will be focused on that variant.

- **Dynamic:** the relevance is computed in the initial case memory, and every time that a new case is learned by the system. It is an incremental weighting method.

The section is divided in an introduction to the Rough Sets theory, the basis concepts of Rough Sets theory and the incorporation of Rough Sets into the Case-Based Reasoning System.

3.1 Rough Sets Theory

Zdzislaw Pawlak introduced Rough Sets theory in 1982 [14, 15, 20]. The idea of the Rough Sets consists of the approximation of a set by a pair of sets, called the lower and the upper approximation of this set. In fact, these approximations are inner and closure operations in a certain topology generated by the available data about elements of the set.

The nature of Rough Sets theory made them useful for reducing the knowledge, extracting dependencies in knowledge, reasoning about knowledge, pattern recognition, etc.

We use Rough Sets theory for reducing and extracting the dependencies in the knowledge. These dependencies are the basis for computing the relevance of each feature into the Case-Based Classifier System.

3.2 Rough Sets inside Case Based Reasoning System

How Rough Sets theory is incorporated into our Case-Based Classifier System?

First of all, we incorporate some concepts in this paper to explain how the dependencies we are looking forward from the domain are obtained to select the best weighting.

3.2.1 Basic Concepts and Definitions

We compute from our **Universe (U)** (finite set and not null set of objects that describes our problem, the case memory) the **concepts** (objects or cases) that form partitions of that Universe. The union of all the concepts make the entire Universe. Using all the concepts we can describe all the **equivalence relations (R)** over the universe (U). Let an equivalence relation be a set of features that describe a specific concept. U/R are the family of all equivalence classes of (R).

The universe and the relations form the **knowledge base (KB)**, defined as KB = < U, $\hat{R}$ >. Where $\hat{R}$ is the family of equivalence relations over U. Every relation over the universe is an elementary concept in the knowledge base.

All the concepts are formed by a set of equivalence relations that describe them. Thus, we search for the minimum set of equivalence relations that define the same concept as the initial set.

Definition 1 (Indiscernibility Relations)
It can be defined as IND($\hat{P}$)= $\bigcap \hat{R}$ where $\hat{P} \subseteq \hat{R}$. The indiscernibility relation is the intersection of properties over P. The indiscernibility shows the refined information over a concept and gives all the information about the equivalence relation that exists in $\hat{P}$.

Example 3.1
If we consider a set of 8 objects in our Universe, $U = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$, using as a family of equivalence relations over U:$\hat{R}$ = (P, Q, S). Where P are colours (green, blue, red, yellow); Q are sizes (small, large, medium); and S are shapes (square, round, triangular, rectangular).

U/P = ((x_1, x_4, x_5), (x_2, x_8), (x_3),(x_6, x_7))
U/Q =((x_1, x_3, x_5), (x_6), (x_3, x_4, x_7, x_8))
U/S = ((x_1, x_5), (x_6), (x_2, x_7, x_8), (x_3, x_4))

As it can be seen, every indiscernibility relation divides the Universe in a different way.

Definition 2 (Basic Knowledge)
The basic knowledge is the family of **all** equivalence classes of the equivalence relation IND($\hat{P}$). The basic knowledge shows all the knowledge associated with the family of equivalence relation P.

Definition 3 (P-basic categories)
P-basic categories are those basic properties of the universe, which can be expressed using knowledge from P. They are the building blocks of the existing knowledge.

Let $K = (U, \hat{R})$ be a knowledge base.
IND(K) = (IND($\hat{P}$): $0 \neq \hat{P} \subseteq \hat{R}$) is the family of all equivalence relations defined in K.

Definition 4 (Equivalence, generalisation) (and specialisation of knowledge)
Let K i K' be two knowledge bases:

- if IND(K) = IND(K'), it means that K and K' are equivalent.
- if IND(K) $\subset$ IND(K') then the knowledge base K is finer than K', so K' is a generalisation of K.

3.2.2 Rough Sets

Let $X \subseteq U$ and R be an equivalence relation. We will say that:

- X is *R-definable* if X is the union of some *R-basic* categories; otherwise X is *R-undefinable*.
- The *R-definable* sets are those subsets of the universe which can be exactly defined in the knowledge base K, whereas the *R-undefinable* sets cannot be defined in this knowledge base.
- The *R-undefinable* set will be also called *R-rough*.
- The set $X \subseteq U$ will be called *exact* in K if there exists $R \in IND(K)$ such that X is *R-exact*, and X is called to be *rough* in K, if X is *R-rough* for any $R \in IND(K)$.

Approximations of Set This is the main idea of rough sets, approximate a set by other sets. The next definitions will explain this idea.

Suppose a given knowledge base $K =< U, \hat{R} >$. With each subset $X \subseteq U$ and an equivalence relation $R \subseteq IND(K)$ we associate two subsets called:

- Lower approximation
- Upper approximation

DEFINITION 5 (LOWER APPROXIMATION)
The lower approximation, defined as: $\underline{R}X = \bigcup \{ Y \in U/R : Y \subseteq X\}$. The lower approximation is the set of all elements of U which can be certainty classified as elements of X in the knowledge R.

DEFINITION 6 (UPPER APPROXIMATION)
The upper approximation, $\overline{R}X = \bigcup \{ Y \in U/R : X \bigcap Y \neq \emptyset \}$. The upper approximation is the set of elements of U which can be possibly classified as elements of X, employing knowledge R.

DEFINITION 7 (BOUNDARY)
$\overline{R}X - \underline{R}X$ is the boundary $BN_R(X)$. The boundary is the set of elements, which cannot be classified either to X or to $\neg X$ having knowledge R.

Reduct and Core of knowledge Intuitively, a **reduct** of knowledge is its essential part, which suffices to define all concepts occurring in the considered knowledge, whereas the *core* is the most important part of the knowledge.

Let $\hat{R}$ be a family of equivalence relations and let $R \in \hat{R}$. We will say that:

- R is *indispensable* if $IND(\hat{R}) \neq IND(\hat{R} - R)$; otherwise it is *dispensable*.
- The family $\hat{R}$ is *independent* if each $R \in \hat{R}$ is *indispensable* in R; otherwise it is *dependent*.

DEFINITION 8 (REDUCT)
$\hat{Q} \in \hat{R}$ is a reduct of $\hat{R}$ if :

1. $\hat{Q}$ is *independent*.
2. $IND(\hat{Q}) = IND(\hat{R})$. Using Q it is possible approximate the same as using R.

DEFINITION 9 (CORE)
The set of all indispensable relations in R will be called the *core* of R, and will be denoted $CORE(R)$.

$$CORE(\hat{R}) = \bigcap RED(\hat{R}) \quad (5)$$

where $RED(\hat{R})$ is the family of all reducts of R.

EXAMPLE 3.2
We continue using the example 3.1 to find the reducts and the core of the knowledge. Our equivalence classes are:
$U/P = ((x_1, x_4, x_5), (x_2, x_8), (x_3), (x_6, x_7))$
$U/Q = ((x_1, x_3, x_5), (x_6), (x_3, x_4, x_7, x_8))$
$U/S = ((x_1, x_5), (x_6), (x_2, x_7, x_8), (x_3, x_4))$

Thus the relation $IND(R)$ has the equivalence classes:

$U/IND(\hat{R}) = ((x_1, x_5), (x_2, x_8), (x_3), (x_4), (x_6), (x_7))$

The relation P is indispensable in R, since:

$U/IND(\hat{R} - P) = ((x_1, x_5), (x_2, x_7, x_8), (x_3), (x_4), (x_6)) \neq U/IND(\hat{R})$.

$U/IND(\hat{R} - Q) = ((x_1, x_5), (x_2, x_8), (x_3), (x_4), (x_6), (x_7)) = U/IND(\hat{R})$.

The information obtained is equal, so the relation Q is dispensable in R.

$U/IND(\hat{R} - S) = ((x_1, x_5), (x_2, x_8), (x_3), (x_4), (x_6), (x_7)) = U/IND(\hat{R})$.

Hence the relation S is also dispensable in R.

That means that the classification defined by the set of three equivalence relations P, Q and S is the same as the classification defined by relation P and Q or P and S.

So the reducts and the core are:

$RED(\hat{R}) = ((P,Q), (P,S))$
$CORE(\hat{R}) = (P)$

3.2.3 How introduce the RS in our CBR system?

We can use the information of reducts and the core to weigh the relevance of each feature in the system. An attribute that does not appear in the reducts has a feature weight value of 0.0, whereas a feature that appears in the core has a feature weight value of 1.0. The rest of attributes has a feature weight value depending on the proportional appearance in the reducts. This is the weight feature information used in the Case-Based Classifier System.

Figure 5 shows the meta-level process when the Rough Sets are incorporated into the CBR system. Rough Sets are divided in three steps: the first one discretises the examples, it is necessary to find the most relevant information using the Rough Sets theory; the second step searches the reducts and the core of knowledge using the Rough Sets theory; and finally, the third step uses the core and the reducts of knowledge to decide the feature relevance value.

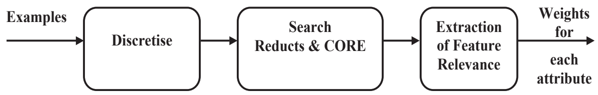

Figure 5: High level process of *Rough Sets.*

The RS theory has been introduced as weighting method in two phases modified of the CBR cycle. The first phase modified with Rough Sets is the *start-up* phase and the second one is the *retain* phase. The start-up phase compute the weights from the initial case memory, these weights will be used by the retrieval phase later. The retain phase computes the weights from the case memory whether the new case is stored and the system works dynamically. The code of Rough Sets theory into the Case-Based Reasoning has been implemented using a public Rough Sets Library [8].

4 Testbed

The experiment has based on 3 data sets from the UCI repository (echocardiogram, iris, breast cancer Wisconsin), and one data set from our own repository (mammogram problem). See table 1 and table 2 which show their characteristics. The mammogram problem consists of detecting breast cancer using the information found in a mammography [12, 13, 17]. A microcalcification (μCa) usually appears, in the mammographies, as small, bright, arbitrarily shaped regions on the large variety of breast texture background. Thus their analysis and characterisation are performed throughout the extraction of features and visibility descriptors by means of several image processing techniques [19]. Each example contains the description of several μCa present in the image. For each of these microcalcifications there are 23 real valued features. In other words, the input information used is a set of $m \times 23$ real valued matrixes, where m is the number of μCa present on the image. The data set contains 216 examples.

The examples of each data set have been grouped in two sets: the training set and the test set. We use the first one to train the system, and the second to test the system. The training set and the test set are generated using different proportions of the examples: 10% of the examples for the training set and the rest (90%) for the test set, 20% of the examples for the training set and the rest (80%) for the test set, ..., until 90% for the training set and 10% for the test set.

Table 1: Data set used for these experiments.

Domain	**Reference**
Echocardiogram	E
Iris	I
Breast cancer (Wisconsin)	BC
Mammogram problem	M

Table 2: Characteristics of the data set used in these experiments.

Ref	**Samples**	**Features**	**Classes**	**Missing Values**	**Inconsistent**
E	132	9	2	132	Yes
I	150	4	3	0	No
BC	699	9	2	9	Yes
M	216	23	2	0	Yes

We have test each data set using the following policies:

- **Similarity Functions:** Minkowski's metric (Hamming, Euclidean and Cubic distance), Clark's distance and Cosine distance.
- **Retain Policies:** DifSim, DifClass and Test.
- **Training initial data set:** training the initial case memory and maintaining the initial case memory.
- **Samples:** we have 9 proportions of each sample and 10 versions for each proportion.

For each data set is tested a total of 2700 runs.

5 Results

We present in this section the main results obtained for each data set tested. Table 3 presents the results obtained during the execution of the proportion 90% training set and 10% test set. The first column is the results obtained using BASTIAN without weighting the attributes, the second column shows the results for the BASTIAN system using the Rough Sets theory as a weighting method. This proportion has been chosen for the accurate rate obtained, we want to notice that the results presented are the maximum value obtained during one run.

Table 3: Maximum results obtained for each data set.

Ref	¬W	RS-W
E	78.57%	78.57%
I	100%	100%
BC	98.71%	98.71%
M	77.27%	81.81%

The results presented obtain a good accuracy rate. We want to outline that the maximum accuracy percentage obtained, using the Rough Sets as a weighting method, appears more frequently than the results obtained without weighting the features.

Figure 6 shows the results obtained for all the training sets proportions in the mammogram problem. As it can be seen, the weighting feature methods needs a huge amount of cases to develop a good weighting for the retrieval phase. However, the system accuracy rate increases when there are enough information in the system to develop a good weighting criterion. Also, the system decreases the standard deviation value if it uses the Rough Sets theory as a weighting method.

We can also notice that it is very important to select a good training of the initial case memory to achieve better results. Thus, most of the best results obtained have been achieved using an initial training. The training set has been decreased following this method. So, the cases chosen were the more representatives to explain the problem.

Table 4 shows the results obtained in different training sets proportions for the Iris problem. The results presented are the maximum and the mean values. As it can be seen there are few differences between the Rough Sets hybrid system and the original Case-Based Classifier System. The results denote also that it is very important the number of cases included into the case memory to achieve a good accuracy in the weighting method.

It is important to remark that the prediction accuracy depends on the case memory size. This fact can be seen in all the problems.

Figure 7 shows the mean results obtained for the echocardiogram problem in all the training set proportions. It also 7 denotes how important is the number of cases into the case memory, and we can also observe that the results depend on the number of missing values.

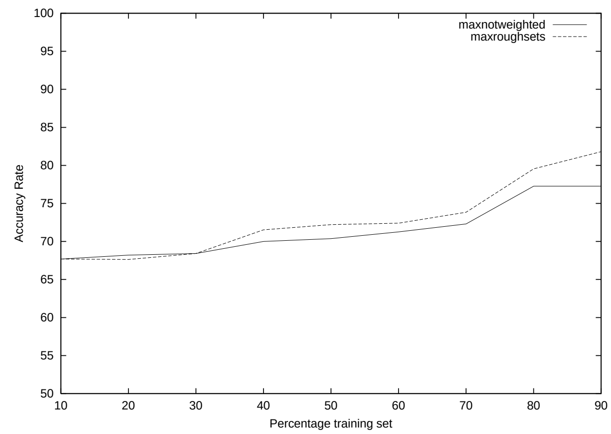

Figure 6: Maximum results obtained in the Mammogram problem.

Table 4: Results for the Iris problem.

Prop. train	Max ¬W	Max RS-W	Mean ¬W	Mean RS-W
40%	98,88	97,77	96,22	96,00
60%	97,77	97,77	95,33	95,50
70%	100,00	100,00	95,11	95,33
80%	100,00	100,00	97,00	97,00
90%	100,00	100,00	96,66	96,66

The results obtained for the Breast Cancer Wisconsin problem can be found in the figure 8. The results are very similar, it is due to the great number of examples in this data set and to the data missing.

Finally, it is important to denote that all the discretisation has been done using the same parameter. This parameter must be changed depending on the upper and lower bounds of each feature. This discretisation influences the results.

6 Conclusions and further work

This paper has proposed how to introduce the Rough Sets theory into a Case-Based Classifier Sys-

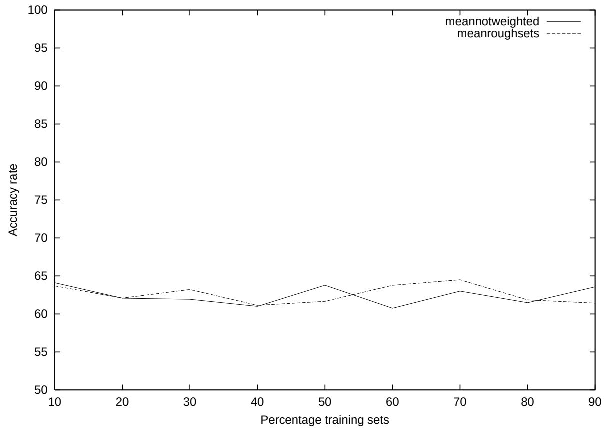

Figure 7: Mean results obtained for the echocardiogram problem.

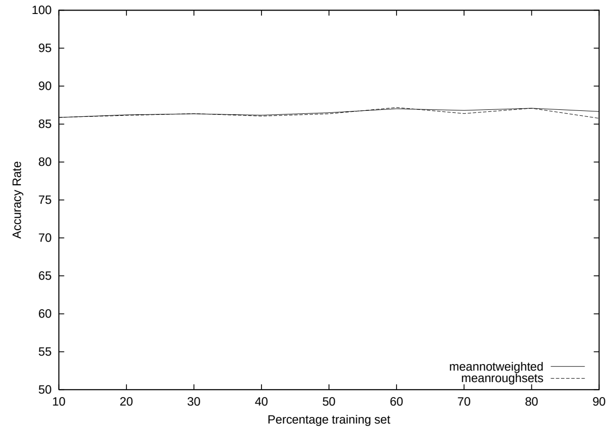

Figure 8: Mean results obtained in Breast Cancer Wisconsin.

tem as a weighting method. The work related here deals with two main ideas: proposing a platform that incorporate Case-Based Reasoning System and the Rough Sets into BASTIAN, and improving the feature relevance mechanism.

We have tested our feature relevance mechanism with different data set from the UCI repository. We have notice that the Rough Sets weighting method improves the accuracy rate if there are enough information into the system to extract the feature relevance. However, the system only decrease the accuracy rate if there are less than a 10% of the cases in memory. The Rough Sets methods help the system to balance the results in the system, there are not many differences between all the versions tested.

Our further work in this area will be to achieve better performance using different criteria on weighting methods and improve the platform introducing new functionalities.

Acknowledgements

This work is supported by the *Ministerio de Sanidad y Consumo, Instituto de salud Carlos III*, Fondo de Investigación Sanitaria of Spain, Grant No. 00/0033-02. The results of this work have been obtained using the equipment co-funded by the *Direcció de Recerca de la Generalitat de Catalunya* (D.O.G.C 30/12/1997).

We would thank Enginyeria i Arquitectura La Salle (Ramon Llull University) for their support to our Research Group in Intelligent Systems.

References

[1] A. Aamodt and E. Plaza. Case-Based Reasoning: Foundations Issues, Methodological Variations, and System Approaches. In *AI Communications*, volume 7, pages 39–59, 1994.

[2] D. Aha. Feature weighting for lazy learning algorithms. Technical Report AIC-98-003, Technical Report, Navy Center for Applied Research in AI, Washington, D.C., 1998.

[3] D. Aha and P. Harrison. Case-Based Sonogram Classification. Technical Report AIC-93-041, Technical Report NRL/FR/5510-94-9707, Navy Center for Applied Research in AI, Washington, D.C., October 1994.

[4] K-D. Althoff et al. Induction and Case-Based Reasoning for Classification Tasks. In H.H. Bock, W. Lenski, and M.M. Richter, editors, *Information Systems and Data Analysis, Prospects-Foundations-Applications, Proc. 17th Annual Conference of the GfKl*, pages 3–16. University of Kaiserslautern, Springer Verlag, Berlin-Heidelberg, 1994.

[5] P. Domingos. Context-sensitive feature selection for lazy learners. volume 11, pages 227–253, 1997.

[6] J.M. Garrell, E. Golobardes, E. Bernadó, and X. Llorà. Automatic diagnosis with Genetic Algorithms and Case-Based Reasoning. In *Artificial Intelligence in Engineering*, volume 13, pages 367–362, October 1999. Elsevier Science Ltd., ISSN 0954-1810.

[7] J.M. Garrell i Guiu, E Golobardes i Ribé, E. Bernadó i Mansilla, and X. Llorà i Fàbrega. Automatic Classification of Mamary Biopsy Images with Machine Learning techniques. In *Proceedings of Engineering of Inteligent Systems (EIS'98)*, 1997.

[8] M. Gawry's and J. Sienkiewicz. Rough Set Library user's Manual. Technical Report 00-665, Technical Report, Institute of Computer Science, Warsaw University of Technology, 1993.

[9] E. Golobardes i Ribé and J.M. Garrel i Guiu. Avaluació d'un sistema classificador basat en casos per a la diagnosi de biòpsies de teixit de glàndules mamàries. In *Proceedings de les Jornades d'Inteligència Artifical (JIA'97)*, 1997.

[10] N. Howe and C. Cardie. Examining locally varying weights for nearest neighbor algorithms. In *Proceedings of the Second ICCBR Conference*, pages 456–466. Springer , Providence, 1997.

[11] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, Inc., 1993.

[12] X. Llorà, E. Golobardes, and M. Salamó. Diagnosis of microcalsification using Case-Based and Genetic Algorithms. In *Proceedings of Engineering of Inteligent Systems (EIS2000)*, volume 1, pages 254– 263, 2000.

[13] J. Martí, J. Español, E. Golobardes, J. Freixenet, R. García, and M. Salamó. Classification of microcalcifications in digital mammograms using case-based reasonig. In *International Workshop on digital Mammography*, 2000.

[14] Z. Pawlak. Rough Sets. In *International Journal of Information and Computer Science*, volume 11, 1982.

[15] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, 1991.

[16] C.K. Riesbeck and R.C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Hillsdale, NJ, US, 1989.

[17] M. Salamó, E. Golobardes, J.M. Garrell, and J. Martí. Clasificación de microcalcificaciones usando Razonamiento Basado en Casos. In *III Jornadas de Transferencia Tecnológica de Inteligencia Artificial (TTIA'99)*, Murcia, Spain, November 1999.

[18] M. Salamó, E. Golobardes, D. Vernet, and M. Nieto. Weighting methods for a Case-Based Classifier System. In *LEARNING'00*, Madrid, Spain, October 2000. IEEE.

[19] L. Shen, R.M. Rangayyan, and J.L. Dessautels. Detection and classification of mammographic calcifications. In *In State of the Art in Digital Mammographic Image Analysis*, pages 198–212, 1994.

[20] A. Skowron. Synthesis of Adaptive Decision Systems from Experimental Data. In A. Aamodt and J. Komorowski, editors, *Fith Scandinavian Conference on Artificial Intelligence*, volume 28, Trondheim, Norway, May 1995. IOS Press.

[21] I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers Inc., 1997.